

## **REMARKS**

Claims 1-4, 6-9, 11, 13-19, and 21-23 were pending in the application. Claims 1-4, 6-9, 11, 13-19, and 21-23 stand rejected by the Examiner. Claims 14-19 are cancelled. Claims 1 and 11 have been amended. Claims 24-27 are new. Support for the new and amended claims can be found in the application as filed, for example, on pages 11-13. No new matter has been added.

### ***Specification***

The Examiner objected to the specification, arguing that a "Transport Layer 30" illustrated in Figure 2 is not a transport layer as it includes a link layer and a network layer. The Examiner is referred to the reference numeral 32 identifying the MAC layer 32. The Examiner is also referred to page 16, ll. 7-8, where a UMAC service access point 322 is described as an interface between a transport layer and a MAC layer. Accordingly, one skilled in the art would understand that in the embodiment illustrated in Figure 2, 30 is a transport layer, 32 is a MAC layer, and SAP 322 is an interface between the transport layer 30 and the MAC layer 32. The Applicant respectfully requests that the Examiner withdraw the objection to the specification.

### ***Claim Rejections - 35 U.S.C. §101***

Claim 11 stand rejected under 35 U.S.C. §101 as not falling within one of the four statutory categories of invention.

As amended claim 11 recites a method of classifying data packets for transmission in a communication system" and includes, *inter alia*, "routing the data packet to a connection associated with the connection identifier; and transmitting the data packet in the communication system." Thus, claim 11 is tied to an apparatus for transmitting data packets in a communication system. Accordingly, the Applicant respectfully requests that the Examiner withdraw the rejection of claim 11.

### ***Claim Rejections - 35 U.S.C. 103***

Claims 1-4, stand rejected under 35 U.S.C. §103(a) as being unpatentable over W. Richard Stevens, "UNIX Network Programming", 1990, (hereinafter Stevens) in view of Raphaeli et al (US 20030103521, hereinafter Raphaeli).

As amended, claim 1 recites “applying classifier rules associated with the service access point to the application data received through the service access point to determine if a connection through a lower protocol layer exists for the application data, including: determining if the application data matches matching criteria of a particular classifier rule; and if the application data matches matching criteria of the particular classifier rule, encapsulating the application data into a payload of a transport message and transmitting the transport messages through a connection of the power line communication system identified by a connection identifier of the particular classifier rule.”

That is, classifier rules are applied to the application data received through the service access point to find a connection. The classifier rules include both matching criteria and a connection identifier. Since classifier rules were previously introduced in dependent claims 17-19, the rejection of claim 1 will be addressed in light of the rejections of claims 17-19 as well.

The Examiner apparently identified a 5-tuple in Stevens as the classifier rule. The 5-tuple identifies a protocol, a local address, a local process, a foreign address and a foreign process. See Stevens, p. 269, l. 8. However, these parameters do not include matching criteria and a connection identifier as described above, nor is this 5-tuple applied to application data.

In Stevens, the API focuses on the interface between a process and a network layer. The 5-tuple is associated with a socket descriptor *sockfd* returned by the *socket* system call and is filled in by other system calls, such as with the *connect* system call, or a *bind* system call followed by a *listen* system call. See Stevens, Figure 6.8.

The Examiner apparently identified the application data as data from the application layer of Figure 5.28 of Stevens. The Examiner also identified a socket created by the various system calls as the service access point through which the application data is received. See Office Action dated December 1, 2009, p. 4. Thus, in the Examiner’s interpretation, the socket associated with the various system calls is between the application layer and the transport layer.

To get the application data into the transport layer for transmission, Stevens described commands such as the *send* system command. The *send* system command includes the socket file descriptor *sockfd* and a buffer *buff*. Nowhere in Stevens is there any description of any of this data being compared with or matched with the parameters of the 5-tuple described above. That is, the data in the buffer *buff* is not matched with a protocol, a local address, a local process, a foreign address or a foreign process, cited by the Examiner as the classifier rule.

Note that in claim 1, the application data that is matched to the matching criteria of the classifier rule is the data that is encapsulated for transmission. That is, the application data is not the socket file descriptor *sockfd* as the socket file descriptor *sockfd* is not described as being encapsulated in a transport message.

Furthermore, the Examiner is apparently using the system calls described in Stevens in various locations in a protocol stack. For example, the Examiner identifies a socket as the interface between an application layer and a transport layer. See Office Action, p. 4, ll. 3-8. However, the Examiner then proceeds to cite pseudocode with the socket system calls for a connection from the transport layer to a lower protocol layer. Thus, it is unclear to which socket the Examiner is referring. The Applicant respectfully requests that the Examiner please clarify if the Examiner is referring to the same or different sockets.

Moreover, even if the Examiner is referring to a socket between an application layer and a transport layer, and a socket between the transport layer and a lower level layer, Stevens does not describe the implementation of the transport layer.

The Examiner has referred to the commands listed in Figure 6.1 of Stevens as reciting the internal operation of the protocol layer. See Office Action, p. 16, ll. 8-9. However, Figure 6.1 of Stevens only list various alternative commands for an interface to a protocol layer, not the internal operation of the protocol layer. For example, the *send* system command listed in Figure 6.1 of Stevens recites a socket file descriptor *sockfd* and a buffer *buff* passed as parameters. See Stevens, p. 274. Nowhere in Stevens is there a description of how the buffer is used, how the socket file descriptor is used, or the like. That is, the system commands listed in Figure 6.1 describe the interface, not the implementation. Accordingly, Stevens cannot teach operations such as applying rules to determine a connection.

The addition of Raphaeli to Stevens does not cure the deficiencies of Stevens as described above. Although Raphaeli describes the internal operations of a MAC layer, and described receiving data from a host, there is no teaching of applying classifier rules as recited in claim 1 to such data from a host.

Accordingly, as described above, Stevens and Raphaeli do not teach each and every element of claim 1. The Applicant respectfully requests that the Examiner withdraw the rejection of claims 1 and dependent claims 2-4.

New claims 24-27, dependent on independent claim 1, introduce additional elements not taught by the cited references. Accordingly, claims 24-27 should be allowable.

Claims 6-7, 9 and 21-23 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Andrew S. Tanenbaum, "Computer Networks", Forth edition, 8/9/2002 (hereinafter Tanenbaum) in view of Stevens, further in view of Kurupati, et al. (U.S. Publication No. 2003/0182291, hereinafter Kurupati).

Claim 6, recites "determining an order of rules associated with the classifier to apply to the data packet using a priority of each of the rules, where each rule includes the corresponding priority." That is, each rule includes the priority.

The Examiner noted that Tanenbaum did not teach the determination of the order of the rules. See Office Action, p. 10. The Examiner cited Figure 6.7 of Stevens to teach the rules. See Office Action, p. 11. However, Figure 6.7 is only a list associating various combinations of family, type, and protocol with an actual protocol. Nowhere is there a reference to a priority.

The Examiner is apparently arguing that data transmitted through a stream socket has a higher priority than data transmitted through a datagram socket; however, there is no such teaching in Stevens. Thus, the Examiner is apparently relying on his own knowledge to teach such a feature. The Examiner is reminded that "Any rejection based on assertions that a fact is well-known or is common knowledge in the art without documentary evidence to support the examiner's conclusion should be judiciously applied." MPEP 2144.03. Accordingly, the Applicant demands according to MPEP 2144.03 that the Examiner produce authority to support the assertion that data transmitted through such sockets would be transmitted with the priority described by the Examiner.

The Examiner argues that the function socket() implements rules. See Office Action, p. 11, citing Stevens. However, as described above, Stevens does not teach the internal operations. That is, there is no teaching of what is performed in response to a socket system call other than returned values.

Claim 6 also recites "for each classification parameter of the rule, comparing a field of the data packet identified by a parameter ID of the classification parameter with a value of the classification parameter." That is, there is not only a parameter ID that identifies a field of the data packet received from the applications, but there is also an associated value for comparison.

The Examiner cites Figure 6.7 of Stevens listing various combinations of parameters of a socket system call. However, there is no data packet from an application in a socket system call, there is no indication that there is some ID of where in the data packet the parameters of Figure 6.7 would be, and there is no indication of a value to compare against the data packet.

The Examiner indicated that Tanenbaum and Stevens do not teach using a priority of the rules. See Office Action p. 11. The Examiner cited Kurupati to teach the priority.

However, the priority associated with the rule in Kurupati is for the priority of a packet. “The rule associated with an IP address indicates what action – e.g. ... a priority determination ... is to be taken with respect to a packet having that IP address.” See Kurupati, paragraph [0017], ll. 11-15. See also paragraph [0026] – “to determine a priority of data contained in a packet.”

That is, the priority in Kurupati is a priority of the packet or the data, not a priority of the rule that determines an order the rule is applied. Accordingly, Tanenbaum, Stevens, and Kurupati do not teach each and every element of claim 6. The Applicant respectfully requests that the Examiner withdraw the rejection of claim 6 and dependent claims 7, 9, and 21-23.

With respect to claim 21, the Examiner cites Tanenbaum to apparently teach that a TCP 23 for telnet has a higher priority than a TCP port 25 for SMTP. However, there is no indication in the cited sections that any priority is given to telnet packets over SMTP packets.

Furthermore, even assuming for the sake of argument that telnet packets have a higher priority over SMTP packets, the Examiner is again making the same mistake of confusing priority of packets with priority of rules applied to a packet.

Furthermore, the Examiner identifies an IP destination address and an IP destination port as a parameter ID and a value, respectively, of a classification parameter of a rule. However, as recited in parent claim 6, a parameter ID identifies a field of the data packet for comparison with the value. An IP destination address does not identify where the IP destination port appears in the data packet. For example, an IP destination address of 192.168.1.1 does not identify where in the data packet the IP destination port appears.

Accordingly, Tanenbaum, Stevens, and Kurupati do not teach each and every element of claim 21. The Applicant respectfully requests that the Examiner withdraw the rejection of claim 21 and dependent claims 22 and 23.

Claim 8 stands rejected under 35 U.S.C. §103(a) as being unpatentable over S. Tanenbaum in view of Stevens and Kurupati, further in view of Malkin (U.S. Patent No. 6,272,145 B1).

The addition of Malkin does not cure the deficiencies of Tanenbaum, Stevens, and Kurupati described above with respect to claim 6. For example, Malkin does not address priorities within rules and their usage when applying the rules to data packets. Instead Malkin focuses on multilink communication. See Malkin, Abstract, and col. 1, ll. 6-18. Accordingly, Tanenbaum, Stevens, Kurupati, and Malkin do not teach each and every element of claim 8. The Applicant respectfully requests that the Examiner withdraw the rejection of claim 8.

Claims 11 and 13 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Stevens in view of Hogan, et al. (U.S. Patent No. 4,841,456) hereinafter Hogan.

The Examiner is apparently referencing Figure 6.7 of Stevens on page 268 to teach the rules. However, Figure 6.7 is only a list associating various combinations of family, type, and protocol with an actual protocol. Nowhere is there a reference to a priority.

Moreover, Stevens is silent on what happens in response to the cited *socket* system call. That is, nowhere in Stevens is there a description of the operation of the system calls, only the interface.

Furthermore, claim 11 recites “analyzing an incoming data packet.” The Examiner cited parameters of a *socket* system call such as the family, type, and protocol. However, in the *socket* system call where these are parameters, there is no incoming data packet. In a system call such as a *send* system call, where there is a buffer as described above, only the socket file descriptor is mentioned. There is no mention on how that socket file descriptor or the associated family, type, and protocol parameters are used in conjunction with the *send* system call.

The Examiner referenced Hogan to teach the priority. However, Hogan, addresses rules that are conditions to take an action in a functional test procedure (FTP). See Hogan, col. 7, ll. 36-53. The rules are not sets of parameters for analyzing a data packet. The Examiner notes that the rules of Hogan are “very general and applies to any set of rules.” Office Action, p. 15. However, the example given by Hogan to determine order is using the number of parameters. See Hogan col. 8, ll. 29-31. That is, the rule with the most conditions is evaluated first.

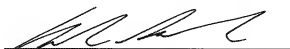
In contrast, the “sets of parameters” in Figure 6.7 of Stevens as cited by the Examiner, each have the same number of parameters. Thus, even assuming that the entries are “sets of parameters”, Hogan provides no indication as to an order of the sets of parameters.

Furthermore, Hogan determines order without using priorities within the sets of parameters. That is, as described above, Hogan merely counts the number of conditions to determine order. Thus, the individual rules in Hogan do not include a priority. In contrast, in claim 1, the order is determined using priorities included in the sets of parameters.

Accordingly, Stevens and Hogan do not teach each and every element of claim 11. The Applicant respectfully requests that the Examiner withdraw the rejection of claim 11 and dependent claim 13.

For the foregoing reasons, reconsideration and allowance of the claims of the application as amended is requested. The Examiner is encouraged to telephone the undersigned at (503) 222-3613 if it appears that an interview would be helpful in advancing the case.

Respectfully submitted,  
MARGER JOHNSON & McCOLLOM, P.C.



Derek Meeker  
Reg. No. 53,313

MARGER JOHNSON & McCOLLOM, P.C.  
210 SW Morrison Street, Suite 400  
Portland, OR 97204  
503-222-3613  
Customer No. 46404